J2EE Design Notes

James Baldo Jr.

SWE 432
Design and Implementation of Software for the Web
N-Tier Architecture

Client-server … 3-tier … N-tier …
Design Goals

• A major design goal of the N-tier architecture is separation of concerns:
  – Presentation
  – Logic
  – Data

• Also to support our seven criteria:
  – Maintainability
  – Security
  – Scalability
  – …
Separation of Concerns

- Presentation → HTML & JSP
- Data content → Servlets & beans
- Data representation → Data structures
  Beans & Java classes
- Data storage → Data base & files
  Oracle, SQL
Separation of Concerns (2)

- `doGet()` and `doPost()` should be simple and short
  - Shift processing to other methods and classes
- Put **complicated logic** in non-servlet classes
- Put almost **no logic** in JSPs
  - JSPs should **present** data they get from other classes

- Use JSPs to **present** data that is on server
- Use servlets to **process** user input
Design Specification

• Software **Requirements** Baseline
• **Information Architecture** Specification
  – Site map, Web Page Flows, Compositions, Labeling, Data Element Mappings
• Web Application **Design**
  – High-Level Software Design
  – Software Architecture and System Architecture Diagram
  – Class Diagrams
  – Sequence Diagrams
  – Class Specifications
Model-View-Controller (MVC)

- The MVC architecture is an abstraction frequently used in web application design
  - Provides a way to divide the responsibilities of objects
  - Decreases coupling between objects and layers (supports easier maintenance)
  - Helps divide the work – supports development expertise areas
Model-View-Controller (MVC)

Model
- Encapsulates application state
- Responds to state queries
- Exposes application functionality
- Notifies views of changes

View
- Renders the model
- Requests updates from the model
- Sends user inputs to controller
- Allows controller to select view

Controller
- Defines application behavior
- Maps user actions to model updates
- Selects a view for response
- One view for each function

* Graphic from Designing Enterprise Applications with the Java 2 Platform, Enterprise Edition, Nicholas Kassem et al., October 2000
Web Application Design Complexity

HTML Pages

Basic JSPs and Servlets

JSPs with Modular Components

JSPs with Modular Components and Enterprise beans

Graphic from Designing Enterprise Applications with the Java 2 Platform, Enterprise Edition, Nicholas Kassem et al., October 2000
Common Design Pitfalls

• No design specifications and no comments in code
• Overly limiting collaboration amongst the development team - only 1-2 people understanding and owning the design
• Coding for future requirements
  – Don’t code ahead
• Using only parts of a documented design framework that are too elaborate
Best Practices

• Establish a Software Requirements Baseline
• Create design specifications (before coding!)
• Use Java Doc
• Teach entire development team the design patterns and design constructs selected for the application, especially the connection points between tiers
  – Every member should be able to explain the design
• Use meaningful names for packages, classes, methods, and variables
  – Ask your teammates if they can understand your names
• Use object-oriented principles to design and develop adaptable systems
  – SWE 619